Hands on New Physics with Machine Learning

Seventh Lisbon Mini-school on Particle and Astroparticle Physics

Guilherme Soares
guilherme.m.s.soares@tecnico.ulisboa.pt

Nuno Leonardo
nuno.leonardo@cern.ch

Laboratory of Instrumentation and Experimental Particle Physics

14 May 2022

Index

Introduction

Goals of this workshop :

- Give an introduction on Machine Learning
- Motivate the use of Machine Learning in Experimental Particle Physics
- Give a small insight on Hidden Sector Physics
- Show a traditional analysis
- Perform a simple Neural Network analysis on SHiP data

You will need a Google account to access Google Colabs !

From here you can run notebooks and try out Machine Learning algorithms using Google's resources.

All relevant files are in this Github repository.

Machine Learning - What is it ?

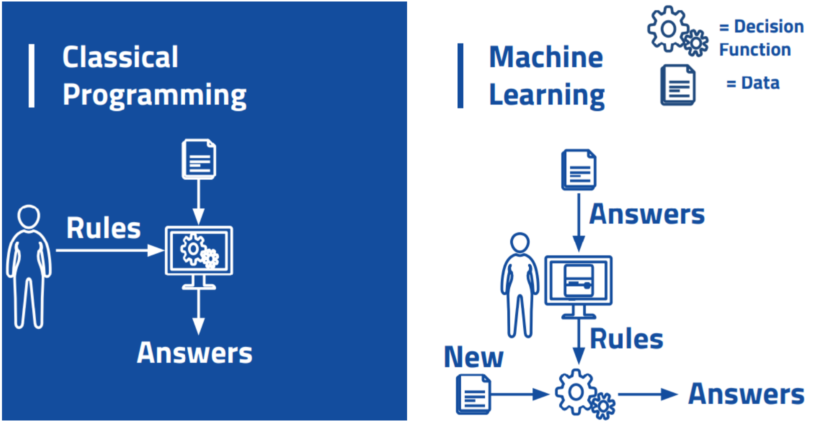


Figure – Miguel Romão, Machine Learning Tutorial, 6th Lisbon Mini-school on Particle and Astroparticle Physics

Machine Learning - Typical Case

How does Machine Learning answer problems ?

- Define suitable Machine Learning structure (e.g. Neural Networks)
- Provide preexisting data. Supervised learning methods need data labeled (for classifiers data X_i is provided as $[X_i, y_i]$, where Y_i is the class)
- Allow the structure to find a near-optimal solution. Iteratively minimizing a loss function is a common solution for classifiers. Binary cross-entropy : $L = \frac{1}{N}\sum_i y_i log(p_i) + (1-y_i) log(1-p_i)$
- Analyze new data with the structure (with its previously found solution)
- Utilize the output to obtain results (e.g. classification problem $\rightarrow$ probability scores p_i)

When are Machine Learning methods viable ?

Good Machine Learning approaches require :

- Large amounts of data
- Accurate data
- **Complex problems** (No exact or trivial solution)

Why use Machine Learning methods in Particle Physics ?

Machine Learning and Experimental Particle Physics : A match made in heaven

The crux of Experimental Particle Physics is to find specific particles or interactions :

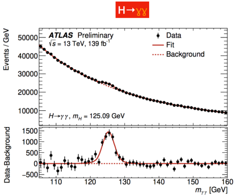
Lots of classification problems

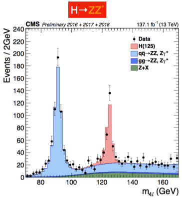
We want to **distinguish background from useful signal**

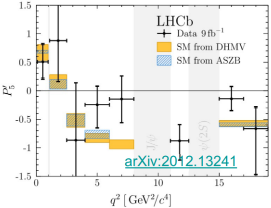
Using simulations to compare distributions with and without signal : **Previously existing accurate data**

Not just looking for excess in distributions, **event selection is key** : **Classification problems**

High amount of variables and features in each event leading to multi-dimensional correlations : **Complex problem**







arXiv:2012.13241

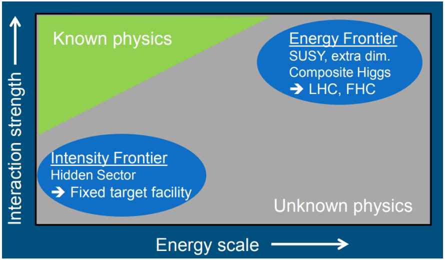
Current State of Particle Physics

The Standard Model is a very successful model, that describes all known elementary particles and their interactions up to the TeV scale. However, it is not able to explain some outstanding phenomena such as :

- Baryonic Assymetry of the Universe (BAU)
- Dark Matter
- Neutrino Oscillations
- Departure from Leptonic Flavour Universality

Can we solve all of this summarily ?

Beyond the Standard Model Physics

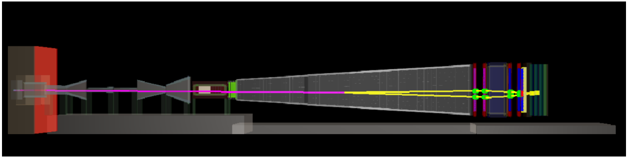


Requirements of Hidden Sector searches :

- High Luminosity
- Long Expected Lifetime
- Controlled Background Levels

The Search for Hidden Particles Experiment

The SHiP Experiment



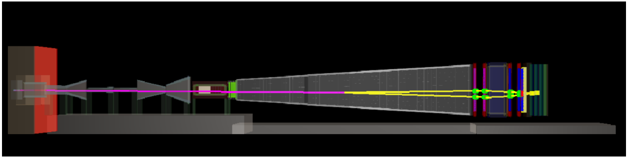
SHiP is a discovery experiment designed to find particles whose production is heavily suppressed ($O(10^{-10})$), with masses of $< \mathcal{O}(10)$GeV/c^2.
Experiment details :

- 400 GeV/c protons
- 2×10^{20} p.o.t.
- 5 years running
- Discoveries through $>$ 2 decays

Therefore, the background must be totally under control to ensure a zero background environment.

The expected relevant sources of background are the following :

The SHiP Experiment



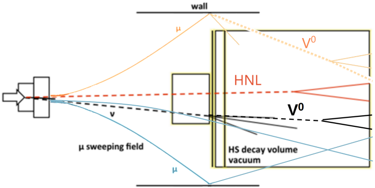
SHiP is a discovery experiment designed to find particles whose production is heavily suppressed ($O(10^{-10})$), with masses of $< \mathcal{O}(10)\text{GeV/c}^2$.
Experiment details :

- 400 GeV/c protons
- 2×10^{20} p.o.t.
- 5 years running
- Discoveries through > 2 decays

Therefore, the background must be totally under control to ensure a zero background environment.

The expected relevant sources of background are the following :

- Neutrino Deep Inelastic Scattering
- Muon Deep Inelastic Scattering
- Muon Combinatorial



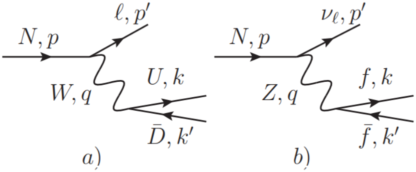
Heavy Neutral Leptons

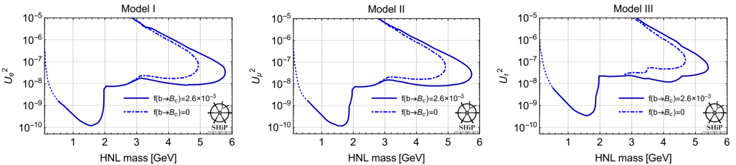
HNLs (N) are massive right-handed **neutrino-like particles** that only couple to the Standard Model through the SM neutrinos (neutrino oscillation-like). HNLs in the Neutrino Minimal Standard Model (**νMSM**) are of especial interest at SHiP.

$$\mathcal{L}_N = \overline{N}_I i\partial_\mu \gamma^\mu N_I - \left[F_{\alpha I} \overline{\ell_\alpha} N_I \tilde{\Phi} + \frac{M_I}{2} \overline{N}_I^c N_I + \text{H.c.} \right]$$

The νMSM contains three HNLs. It **solves** :

- Neutrino masses
- Dark Matter candidate
- Can explain BAU





Distinguishing New Physics from the SM

Goal of the Analysis : Differentiating whether our detectors saw a SM particle (or interaction) or New Physics.

Resources :

- PID
- **Kinematic Features**

Total Momentum
Transverse Momentum
Fraction of Transverse Momentum
Opening Angle
Impact Parameter
Coordinates of the Decay Vertex
Distance Of Closest Approach (DOCA)

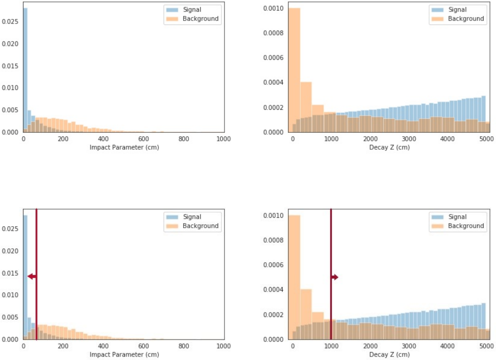
We **simulate Hidden Sector particles** and compare them to what we expect from the Standard Model.

Data Analysis

We want to separate as well as possible our Heavy Neutral Lepton signal from the background processes.

Standard procedure :

- First compare the distributions from their kinematic features.
- Select the most distinguishable ones.
- Define criteria that separates the datasets as well as possible.

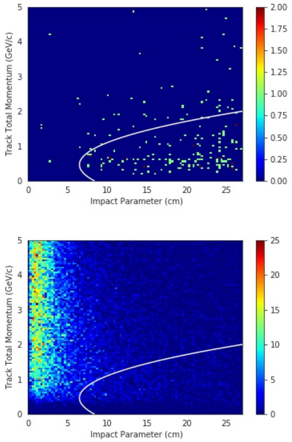


Data Analysis

We want to separate as well as possible our Heavy Neutral Lepton signal from the background processes.

Standard procedure :

- First compare the distributions from their kinematic features.
- Select the most distinguishable ones.
- Define criteria that separates the datasets as well as possible.



Neural Networks - Multi-Layer Perceptrons

Multi-layer Perceptrons : Non-linear feedforward Neural Networks

Neurons are divided in layers.

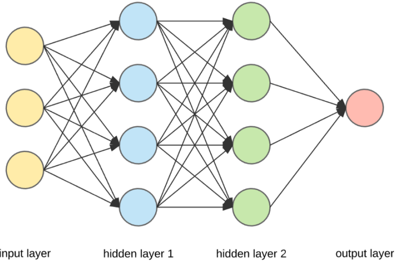
Information is processed and transmitted from layer to layer

Each connection is **weighted**, and the weights are updated iteratively

Provides high processing power with a simple structure : Leads to **physical abstraction**

Since the machine only sees numbers, **features should be normalized** to have the same order of magnitude : $\mu = 0$ and $\sigma = 1$ is a common procedure

Deals very well **with non-trivial correlations between features**



$$\vec{h_1} = a_1 \left(\vec{w_1} \cdot \vec{I} + \vec{b_1} \right)$$

$$\vec{h_2} = a_2 \left(\vec{w_2} \cdot \vec{h_1} + \vec{b_2} \right)$$

$$Out = a_{Out} \left(\vec{w_{Out}} \cdot \vec{h_2} + \vec{b_{Out}} \right)$$

$$a_i = \text{activation function}$$

$$NN = Out \otimes \vec{h_2} \otimes \vec{h_1}$$

Multi-Layer Perceptrons - Model Choice

Defining a good model is most of the process

There is always 1 neuron per feature input in the Input layer

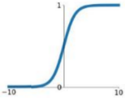
More hidden layers and more neurons per layer imply more processing power. This also requires more computing power : need **simple and effective Activation Functions**

Do bigger networks always lead to better results ?

Overfitting can be problematic. Using a n-polynomial perfectly fits any dataset with n entries, but provides very little insight into the system)

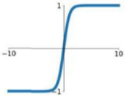
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



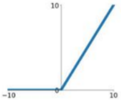
tanh

$\tanh(x)$



ReLU

$\max(0, x)$



Multi-Layer Perceptrons - Quality Insurance

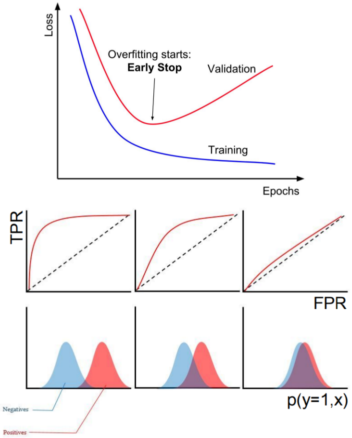
The first step to avoid overfitting is guaranteeing that the model does not update solely based on the events it sees.

We separate the data into 3 samples : **Training**, **Validating** and **Testing**.

Early stopping should be applied in order to choose the best combination of weights.

Our final model is tested with the remaining data (Testing). The **ROC curve** is a good indicator of separation

Due to the inherent complexity and randomness of Machine Learning methods, always vary the parameters of you Neural Network, and train different combinations several times. In the end, choose the best iteration.



Last Thing - DIY

After running through the template with the default settings, try to create a Neural Network that analyses the more difficult sample of HNLs that decay to a rho meson accompanied by a muon.

In order to do this try to use 5 different kinematic features, and vary the relevant parameters of your NN like the depth and width of your hidden layers, and the epochs or batch sizes.

If you feel like you want to spend a bit more time, you can also try to alter the **dropout()** parameters along the hidden layers, to fine-tune the training by removing random information from layer to layer (usually done to avoid overfitting).

Try to obtain the least amount of background events for the testing sample, while maintaining a selection efficiency of the signal at 95% (for the testing sample respectively).